الگوریتمی جدید جهت تولید چندضلعی ساده از مجموعه نقاط مسطح تصادفی به روش ترسیم خطوط غیر متقاطع

علی نوراله، محمد ابوئی مهریزی
دانشکده مهندسی برق و کامپیوتر، دانشگاه تربیت دبیر شهید رجایی، تهران

چکیده

تولید چندضلعی ساده از جمله مسائل مطرح در هندسه محاسباتی می‌باشد. این مسئله در بررسی صحت عملکرد الگوریتم و نیز در زمینه‌های مختلف گرافیک کامپیوتری نظیر خلق تصاویر پدیده‌های طبیعی تصادفی (مانند پدیده ابر و سطح زمین)، به طور گسترده‌ای مورد استفاده قرار می‌گیرد. در این مقاله الگوریتمی جدید جهت تولید چندضلعی ساده از یک مجموعه نقاط تصادفی واقع بر صفحه ارائه شده است. در این الگوریتم خطوطی باشرایط خاص ترسیم می‌شوند و با توجه به متقاطع بودن یا نبودن نسبت به خطوط قبلی، عملیات مناسب جهت رسم پاره خط مورد نظر اعمال می‌شود. پیچیدگی زمانی این الگوریتم از مرتبه $O(nlogn)$ می‌باشد که نشان می‌دهد الگوریتم از نظر زمانی بهینه می‌باشد. از آنجایی که زمان اجرا در برخی از کاربردهای بلادرنگ نظیر بازی‌های رایانه‌ای از اهمیت بالایی برخوردار است، این الگوریتم میتواند در این موارد بسیار مفید واقع شود.

کلید واژه

چندضلعی تصادفی، گرافیک کامپیوتری، هندسه محاسباتی.

۱- مقدمه

تولید چندضلعی‌ها در مسائل مختلفی از جمله گرافیک کامپیوتری و آزمایش صحت الگوریتم‌ها کاربرد دارد [1]. خلق صحنه‌های انیمیشن در بازی‌ها و سرگرمی‌های رایانه‌ای و تولید تصاویر بافت‌های پدیده‌های طبیعی نظیر ابرها و سطح زمین از نمونه‌های کاربرد مساله تولید چندضلعی در گرافیک کامپیوتر می‌باشند. به منظور آزمایش دقیق صحت عملکرد الگوریتم‌ها، نمونه‌های ورودی مختلف و تصادفی می‌بایست مورد آزمون واقع شوند. درنتیجه تولید چندضلعی‌ها در فراهم آوردن تنوع مطلوب نمونه‌های ورودی، کاربرد بسیار دارد.

روش‌های مختلفی برای تولید چندضلعی از مجموعه نقاط مسطح بیان شده‌اند. در [2] الگوریتم‌های اکتشافی جهت تولید چندضلعی‌های خاص، نظیر چندضلعی‌های ستاره‌ای شکل و یکنوا ارائه شده است.

از آنجا که تولید چندضلعی از مجموعه نقاط تصادفی در بررسی عملکرد الگوریتم‌ها بسیار مورد استفاده قرار می‌گیرند، لذا در برخی از کاربردهای بلادرنگ (نظیر بازی‌های رایانه‌ای) زمان اجرای الگوریتم‌ها، اهمیت بسیار زیادی دارد. در این مقاله، الگوریتمی ارائه شده است که چندضلعی را با استفاده از مرتب سازی در زمان $O(nlogn)$ تولید می‌کند.

به منظور بیان مطالب بعدی و الگوریتم پیشنهادی، در بخش دوم تعاریف مورد نیاز ارائه می‌شوند. در بخش سوم کارهای انجام شده و در بخش چهارم الگوریتم پیشنهادی بیان می‌شود، در بخش پنجم تحلیل زمانی و در بخش ششم جمع‌بندی مطالب بیان شده ارئه خواهد شد.

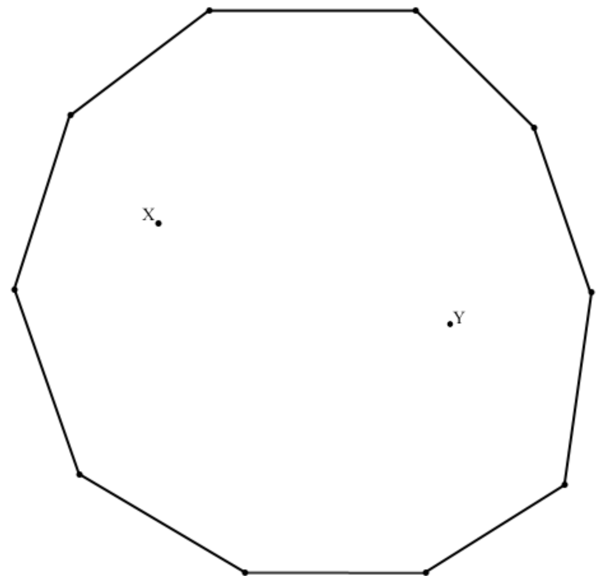
۲- تعاریف اولیه

فرض کنید مجموعه S شامل n نقطه $p_1, p_2, \dots, p_n$ در فضای اقلیدسی باشد. هدف مساله تولید چندضلعی، ساخت یک چندضلعی ساده مانند P است به طوریکه مجموعه رئوس چندضلعی تولید شده، فقط شامل عناصر S باشد. نکته‌ای که توجه به آن لازم است، این است که چندضلعی‌های قابل تولید از یک مجموعه نقاط، یکتا نمی‌باشند.

چندضلعی ساده P، ناحیه‌ای در صفحه است که توسط تعداد محدودی پاره‌خط احاطه شده است به طوریکه یک دور ساده ایجاد گردد [3]. منظور از احاطه کردن جدا کردن قسمتی از صفحه با پاره‌خط‌هایی که در امتداد یکدیگر رسم می‌شوند و انتهای آخرین پاره‌خط به ابتدای اولین پاره‌خط وصل می‌شود. در یک چندضلعی ساده اشتراک پاره‌خط‌های (اضلاع) غیر مجاور P تهی است و دو

پاره‌خط (ضلع) مجاور تنها در نقاط پایانی خود (رئوس P) با یکدیگر تقاطع دارند.

در صورتی که تمام زوایای داخلی چندضلعی P کوچکتر از π باشند، P را یک چندضلعی محدب می‌نامیم. اشتراک پاره‌خط واصل بین هر دو نقطه متعلق به چندضلعی محدب و محیط چندضلعی، تهی می‌باشد. به عبارت دیگر این پاره‌خط هیچ یک از اضلاع چندضلعی را قطع نمی‌کند. در چنین شرایطی این دو نقطه از یکدیگر قابل رویت می‌باشند. نقطه x قابل رویت از نقطه y است (نقطه x نقطه y را رویت می‌کند) اگر و تنها اگر هیچ ریز مجموعه‌ای از پاره‌خط xy در خارج چندضلعی واقع نشود. نقطه x را اکیداً قابل رویت از نقطه y می‌نامیم اگر پاره‌خط xy به طور کامل داخل چندضلعی واقع شود (شکل ۱).



شکل ۱: چند ضلعی محدب – X و Y اکیداً قابل رویت هستند.

ناحیه‌ای از چندضلعی که با قرار دادن یک منبع روشنایی در هر نقطه از آن بتوان تمام چندضلعی را روشن نمود، هسته چندضلعی نامیده می‌شود. هسته یک چندضلعی ستاره‌ای شکل غیر تهی است. یک چندضلعی یکنوا در راستای خط L، خط عمود بر L را تنها در دو نقطه قطع می‌نماید.

پوسته محدب مجموعه نقاط d بعدی S ($d \geq 1$)، اشتراک تمامی نیم فضاهایی است که S را دربر می‌گیرند. پوسته محدب مجموعه نقاط S را با $CH(S)$ نشان می‌دهیم. کوچکترین چندضلعی محدب P که S را احاطه نماید، پوسته محدب مجموعه نقاط مسطح S نامیده می‌شود. به عبارت دیگر هیچ چندضلعی دیگری مانند P' وجود نداشته باشد که $S \subseteq P' \subset P$ [3].

۳- کارهای انجام شده

اخیراً تولید اشیاء هندسی تصادفی و مخصوصا چندضلعی ساده مورد توجه محققین قرار گرفته است؛ برای مثال، Epstein تولید تصادفی مثلث‌بندی را مورد مطالعه قرار داد [8]. همچنین در [9] الگوریتمی برای تولید چندضلعی‌های یکنواخت بر اساس مختصه x (*x-monoton*) بر روی مجموعه نقاط تصادفی ارائه شده است. الگوریتم دیگری جهت تولید چندضلعی ساده از O'Rourke و Virmani ارائه شده است [10]. Auer و Held الگوریتم‌های زیر را ارائه کرده‌اند [2]:

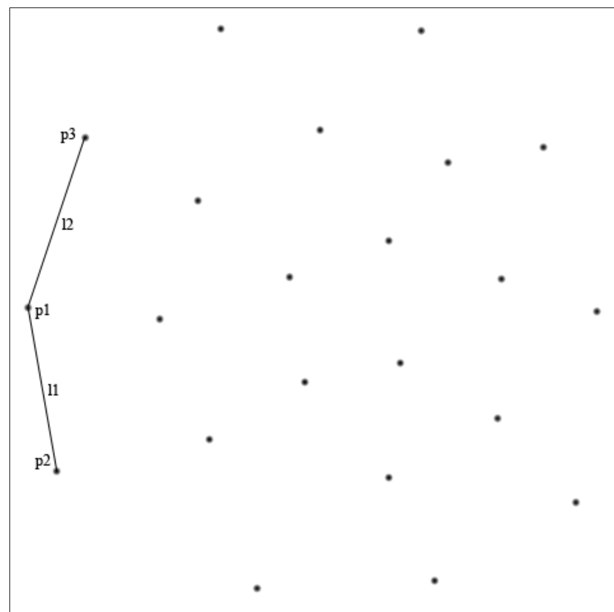
- رشد مداوم (Steady Growth): یک الگوریتم افزایشی است که نقاط را یکی پس از دیگری اضافه می‌کند؛ پیچیدگی زمانی این الگوریتم در بدترین حالت $O(n^2)$ و در بهترین حالت $O(nlogn)$ می‌باشد.
- بخش‌بندی فضا (Space Partitioning): این الگوریتم که از نوع تقسیم و غلبه می‌باشد دارای مرتبه زمانی $O(n^2)$ است.
- تغییر دادن و رد کردن (Permute & Reject): این روش تغییرات تصادفی ایجاد می‌کند، سپس تصمیم می‌گیرد که آن به چندضلعی تصادفی ختم می‌شود یا خیر، تا زمانی که چندضلعی ساده بدست آید. مرتبه زمانی این الگوریتم $O(nlogn)$ می‌باشد.
- انتقال دو انتخابی (2-opt Moves): این الگوریتم از یک چندضلعی تصادفی شروع می‌کند و لبه‌های متقاطع آنرا جایگزین می‌کند تا چندضلعی ساده بدست آید. مرتبه زمانی این الگوریتم $O(n^4)$ می‌باشد.

۴- الگوریتم پیشنهادی

در این بخش الگوریتم پیشنهادی برای تولید چندضلعی ارائه می‌گردد نکته قابل توجه اینکه در انجام مراحل الگوریتم فقط بر چهار نقطه از مجموعه نقاط ورودی تمرکز داریم و بعد از تعیین شرایط نقطه چهارم نسبت به سه نقطه قبل سراغ چهار نقطه بعدی می‌رویم که سه‌تای آنها با چهار نقطه قبلی مشترک بوده و فقط یک نقطه به آنها اضافه می‌شود.

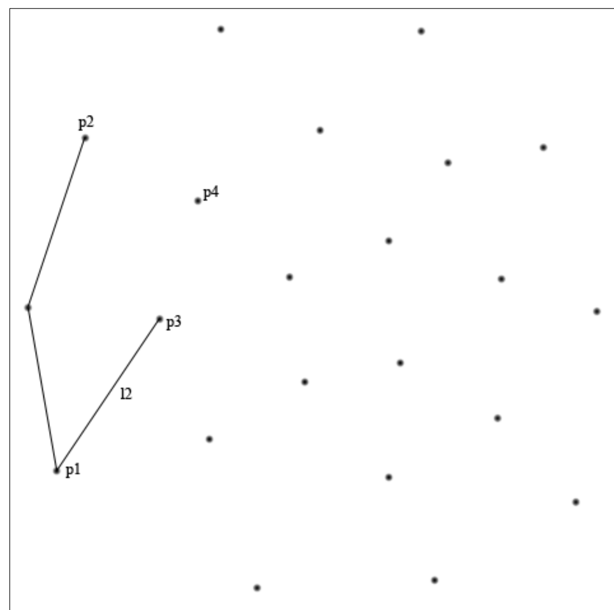
مرحله اول: نقاط را بر اساس مختصه Xشان بصورت صعودی مرتب می‌کنیم [4, 5, 6, 7] و آنها را $p_1, p_2, \dots, p_n$ می‌نامیم.

مرحله دوم: p_1 را به p_2 وصل می‌کنیم (l_1)؛ همچنین p_1 را به p_3 نیز وصل می‌کنیم (l_2) (شکل ۲).



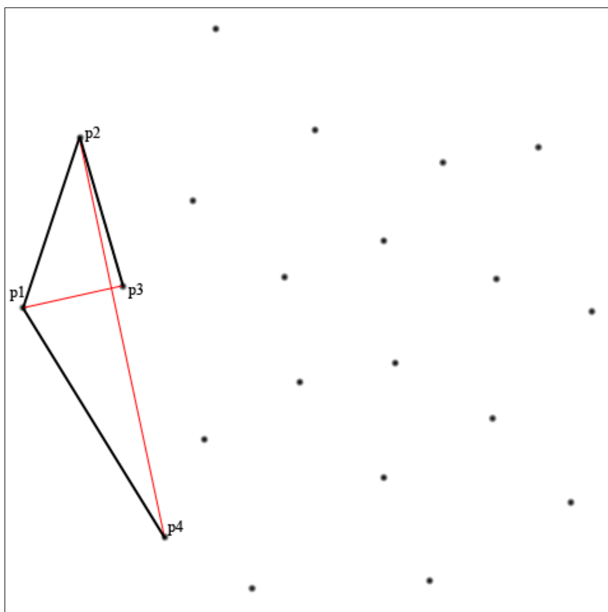
شکل ۲: اتصال p_1 به p_2 و p_3

مرحله سوم: سپس p_2 را به p_4 وصل می‌کنیم، اگر l_2 را قطع نکرد همین عملیات را برای نقاط بعدی بدینصورت انجام می‌دهیم که p_2 را بعنوان p_1، p_3 را بعنوان p_2، p_4 را بعنوان p_3 و نقطه بعدی را بعنوان p_4 در نظر گرفته و مرحله سوم را تکرار می‌کنیم (شکل ۳)؛



شکل ۳: اتصال p_2 به p_4 و عدم وجود تقاطع با l_2

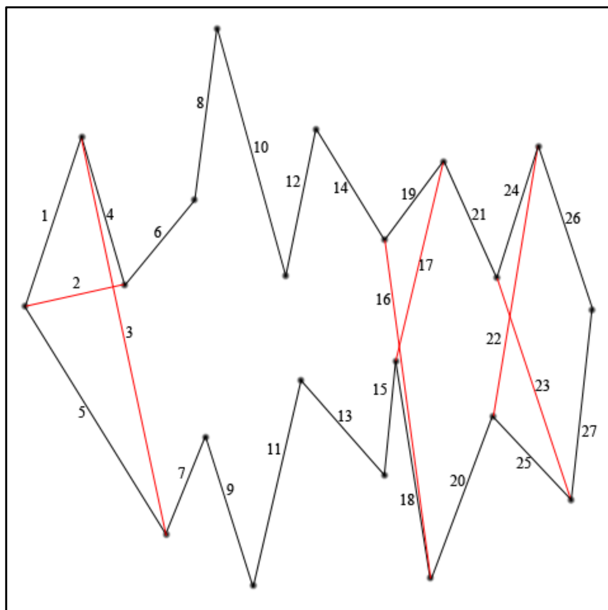
ولی اگر l_2 را قطع کرد، هر دو خط l_1 و l_2 را حذف کرده و نقطه p_1 به p_4 و p_2 به p_3 وصل می‌شوند (شکل ۴) و همین عملیات با این تفاوت که p_3 را بعنوان p_2 و p_4 را بعنوان p_3 در نظر می‌گیریم تکرار می‌کنیم.



شکل ۴: وجود تقاطع بعد از اتصال p_2 به p_4 و حذف هر دو خط l_1 و l_2 و ترسیم خطوط صحیح

مرحله چهارم: در نهایت دو نقطه آخر را بهم وصل کرده و چند ضلعی مورد نظر بدست می‌آید.

در زیر تعدادی نقطه تصادفی و شکل نهایی چندضلعی با تعیین ترتیب رسم خطوط نشان داده شده است (شکل ۵).



شکل ۵: شکل نهایی چند ضلعی با نقاط تصادفی

در زیر شبه کد پیشنهادی جهت پیاده سازی الگوریتم مذکور آورده شده است.

Algorithm: polygon generation

Input: a set of flat points.

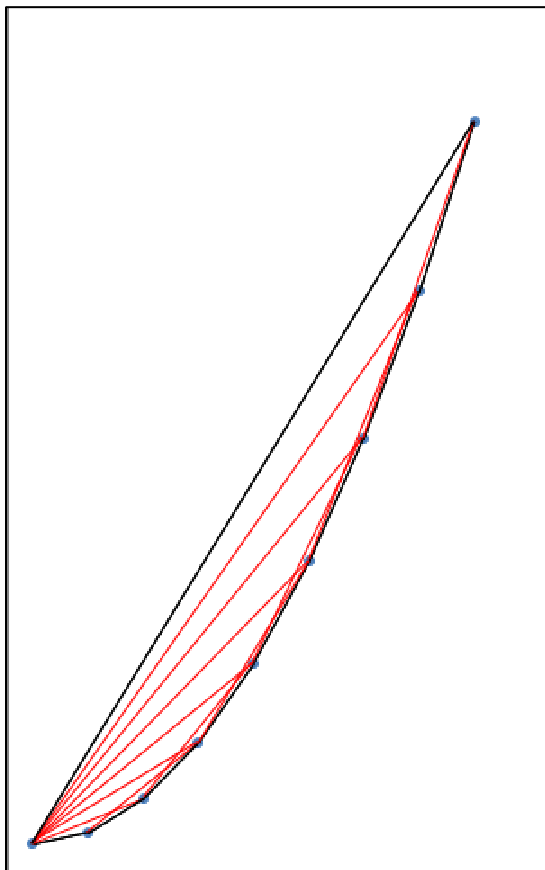
Output: a polygon that its vertices are inputted points.

```
Sort the given points according to X coordinate and call them P_1, P_2, … , P_n
FP ← P_1
SP ← P_2
TP ← P_3
NewP ← P_4
Line(FP, SP)
L_1 ← Line(FP, TP)
Counter ← 3
While Counter < |Points| do
Begin
    Counter ← Counter + 1
    If line(SP, NewP) is crossed L_1 Then
    Begin
        Remove L_1
        L_1 ← Line(FP, NewP )
        Line(SP , TP)
    End
    Else Begin
        Line(SP , NewP)
        FP ← SP
    End If
    SP ← TP
    TP ← NewP
    NewP ← P_Counter
End While

Line(P_{n−1}, P_n)
```

۵- تحلیل پیچیدگی زمانی

بخش مرتب سازی دارای مرتبه زمانی $O(nlogn)$ می‌باشد؛ مراحل بعدی نیز بدین‌صورت خواهد بود که در بدترین حالت نقاط بر روی نمودار سهمی $y = x^2$ $(x > 0)$ قرار دارند (شکل ۶) و بازای هر خطی به یکبار تصحیح جایگاه خطوط نیاز داریم که در این حالت عملیات مربوطه دارای پیچیدگی $O(n)$ می‌باشد؛ در کل پیچیدگی زمانی این الگوریتم $O(nlogn)$ خواهد بود و از آنجا که کمترین زمان برای تولید چند ضلعی $O(nlogn)$ است، لذا الگوریتم از نظر مرتبه زمانی بهینه می‌باشد.



شکل ۶: نقاط روی نمودار سهمی قرار دارند و نقطه اول به هر نقطه ای که متصل می شود باید تصحیح شود.

۶- نتیجه گیری

ما در این مقاله الگوریتمی ابتکاری جدید، ساده و در عین حال کاربردی، جهت تولید چندضلعی ساده از مجموعه نقاط مسطح ارائه کردیم. در این الگوریتم خطوطی با شرایط بیان شده ترسیم می‌شوند و با توجه به متقاطع بودن یا نبودن نسبت به خطوط قبلی، عملیات مربوطه جهت رسم پاره خط مورد نظر اعمال می‌شود. پیچیدگی زمانی این الگوریتم $O(nlogn)$ می‌باشد، لذا الگوریتم از نظر مرتبه زمانی نیز بهینه می‌باشد.

از جمله کاربردهای این الگوریتم می‌توان به بررسی صحت عملکرد الگوریتم‌ها و نیز زمینه‌های مختلف گرافیک کامپیوتری نظیر خلق تصاویر پدیده‌های طبیعی تصادفی (مانند پدیده ابر و سطح زمین) اشاره کرد.

در این الگوریتم فرض بر این بود که مختصات نقاط ورودی قبل از اجرا معلوم است و به عنوان ورودی به الگوریتم داده می‌شود. در آینده الگوریتم پیشنهادی برای حالتی که نقاط در هنگام اجرا و به صورت پویا تولید شوند، توسعه داده خواهد شد.

مراجع:

[1] Dailey D., Whitfield D., "**Constructing Random Polygons**", *SIGITE'08, USA*, pp. 119-124, 2008.

[2] Auer T., Held M., "**Heuristics for the Generation of Random Polygons**", *8th Canadian Conference on Computational Geometry (CCCG)*, pp. 38-44, 1996.

[3] O'Rourke J., "**Computational Geometry in C**", 2nd edition, Cambridge University, 1997.

[4] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. "**Introduction to Algorithms**". MIT Press, Cambridge MA, second edition, 2001.

[5] U. Manber. "**Introduction to Algorithms**". Addison-Wesley, Reading MA, 1989.

[6] Steven S. Skiena, "**The Algorithm Design Manual**", 2nd edition, University of New York, 2008.

[7] J. Kleinberg and E. Tardos. "**Algorithm Design**". Addison Wesley, 2006.

[8] Michael, T.S. "**How to Guard an Art Gallery and other Discerete Mathematical Adventures**"; the Book of the Johns Hobkins University Press, printed in the United Stetes of American, 2009.

[9] Zhu, C., Sundaram, G., Soneyink, J., Mitchell, J.S.B. "**Generating Random Polygon with Given Vertices**"; Comput. Geom. Theory and Appl., Vol 6, Issues 5, 277- 290, 1996.

[10] O'rourke, J., Virmani, M. "**Generating Random Polygons**"; *Technical Report 011, CS Dept. Smith Colege, Northhampton*, MA 01063 1991.